Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.04– «Программная инженерия»

**Лабораторная работа №2**

**«Численные методы решения уравнений»**

**Вариант 22**

Выполнил студент гр. РИС-24-1б

Лесюков Глеб Вадимович

Проверил:

Доц. каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2024

**Постановка задачи:** Решить уравнение методом ньютона.

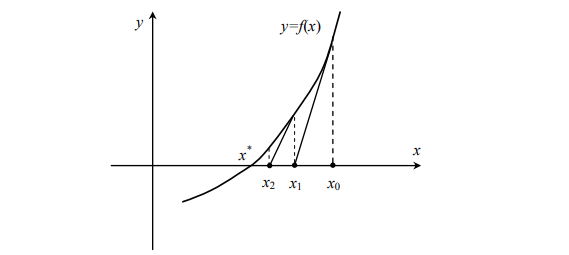
Дано:

F(x)=ex - e-x -2 = 0

Отрезок, содержащий корень: [0;1]

Точное значение: 0,8814

**Геометрическая интерпретация метода:**



**Обоснование стороны подхода к функции:**

Нужно выбирать ту сторону, для которой будет истинно данное неравенство:



**Вывод формулы нахождения корня:**

Выберем на отрезке[0; 1] произвольную точку х0. Затем найдем:

,

потом 

Таким образом, процесс нахождения корня уравнения сводится к вычислению чисел xn по формуле:



Процесс вычисления продолжается до тех пор, пока не будет выполнено условие:



**Блок-схема:**

**Код:**

#include <iostream>

#include <cmath>

using namespace std;

double ddy(double x)

{

return exp(x) - exp(-x);

}

double dy(double x)

{

return exp(x) + exp(-x);

}

double y(double x)

{

return exp(x) - exp(-x)-2;

}

double result\_x1(double x0)

{

return x0-(y(x0)/dy(x0));

}

int main()

{

setlocale(LC\_ALL, "russian");

double epsilon, a, b,x1,x0;

cout << "ВВедите эпсилон, начало интервала и конец интервала: ";

cin >> epsilon>> a>> b;

x0 = (ddy(b)\*y(b)>0)? b: a;

x1 = result\_x1(x0);

while (abs(x1 - x0) >= epsilon)

{

x0 = x1;

x1 = result\_x1(x0);

}

cout << "Корень на интервале: " << x1;

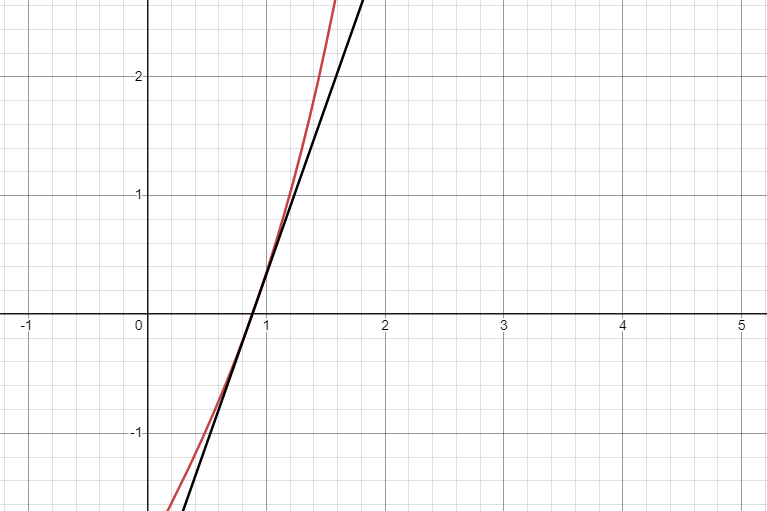
return 1;

}

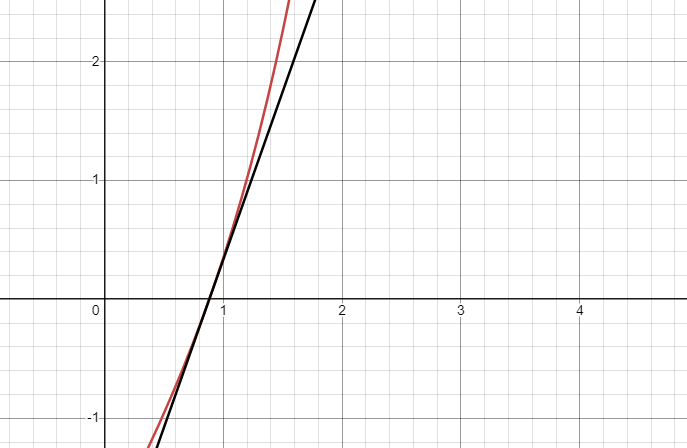
**Пошаговые результаты:**

F(x)-красный, *xn*-чёрный

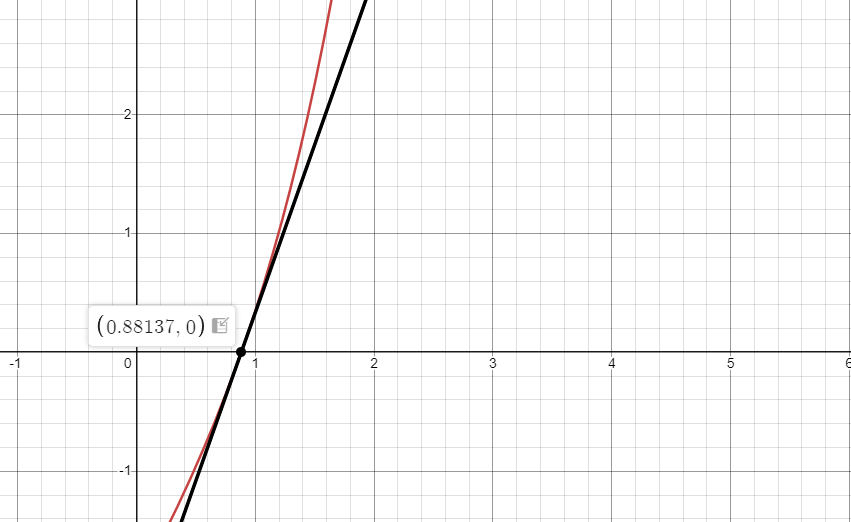
**X=** **0.88646**



**X=** **0.881383**

****

**X=** **0.881374**

****

**Постановка задачи:** Решить уравнение методом итераций.

Дано:

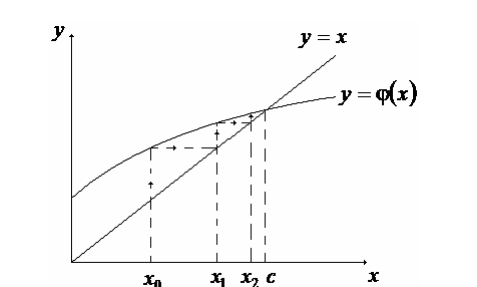
F(x)=ex - e-x -2 = 0

Отрезок, содержащий корень: [0;1]

Точное значение: 0,8814

*f(x)=x*

**Геометрическая интерпретация метода:**

**

**Обоснование стороны подхода к функции:**

Нужно выбирать ту сторону, для которой будет истинно данное неравенство:

**Вывод формулы нахождения корня:**

На заданном отрезке [a; b] выберем точку х0 – нулевое приближение – и найдем

*х1 = f(x0),*

потом найдем:

*х2 = f(x1),*

и т.д.

Таким образом, процесс нахождения корня уравнения сводится к последовательному вычислению чисел:

*хn = f(xn-1) n = 1,2,3..... .*

Процесс итераций продолжается до тех пор, пока

*|xn - xn-1|  ε*

**Блок-схема:**

**

**Код:**

#include <iostream>

#include <cmath>

using namespace std;

double result\_x1(double x0)

{

return x0 + (exp(x0) - exp(-x0) - 2) / -3;

}

double def\_result\_x1(double x0)

{

return 1 + (exp(x0) + exp(-x0)) / -3;

}

int main()

{

setlocale(LC\_ALL, "russian");

double epsilon, a, b, x1, x0;

cout << "ВВедите эпсилон, начало интервала и конец интервала: ";

cin >> epsilon >> a >> b;

x0 = (abs(def\_result\_x1(b))<1) ? b : a;

x1 = result\_x1(x0);

while (abs(x1 - x0) >= epsilon)

{

x0 = x1;

x1 = result\_x1(x0);

}

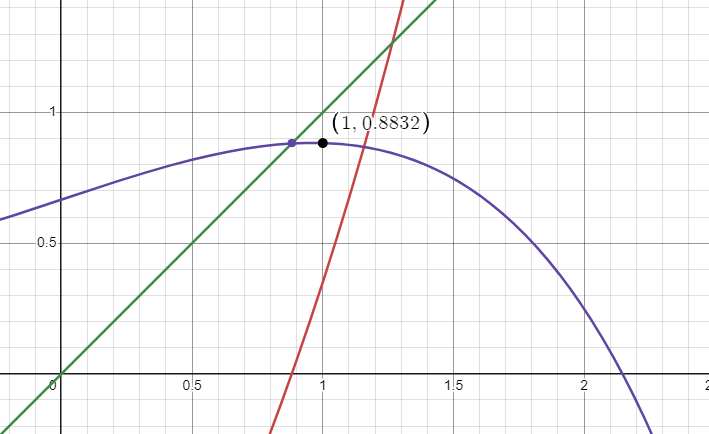
cout << "Корень на интервале: " << x1;

}

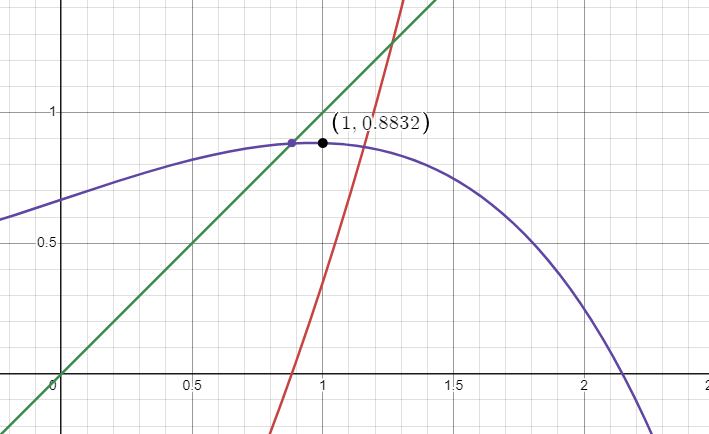
**Пошаговые результаты:**

F(x)-красный, *f(x)*-фиолетовый, y=x -зелёный

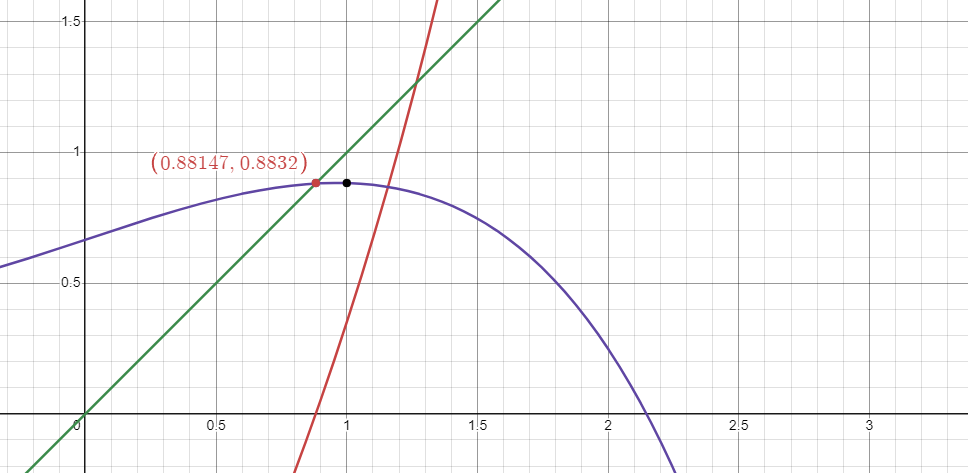
X=1



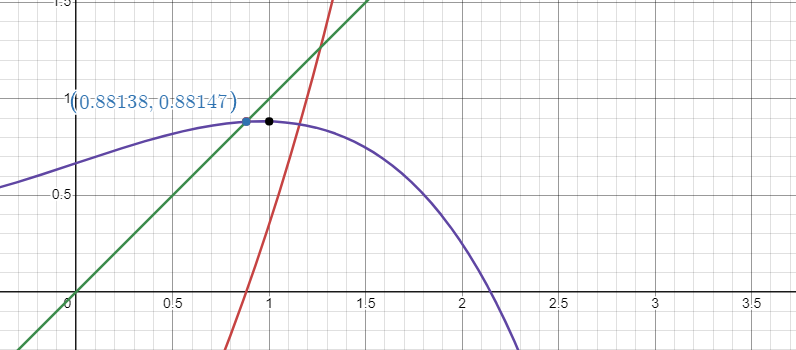
X= 0.8832



X= 0.88147



X= 0.8814



**Постановка задачи:** Решить уравнение методом половинного деления.

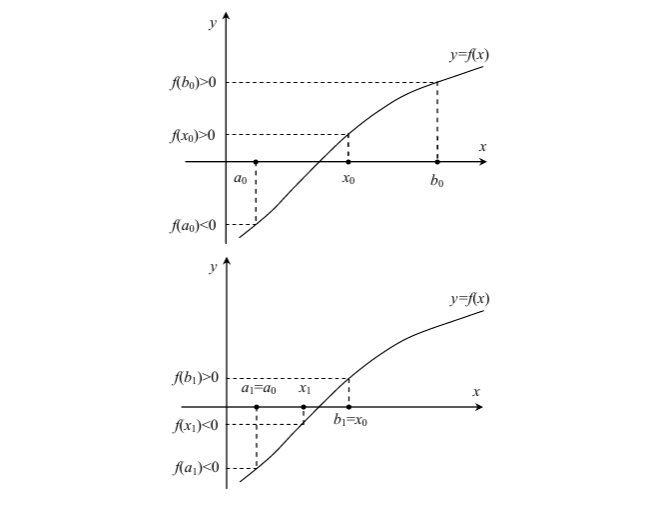
Дано:

F(x)=ex - e-x -2 = 0

Отрезок, содержащий корень: [0;1]

Точное значение: 0,8814

**Геометрическая интерпретация метода:**

****

**Вывод формулы нахождения корня:**

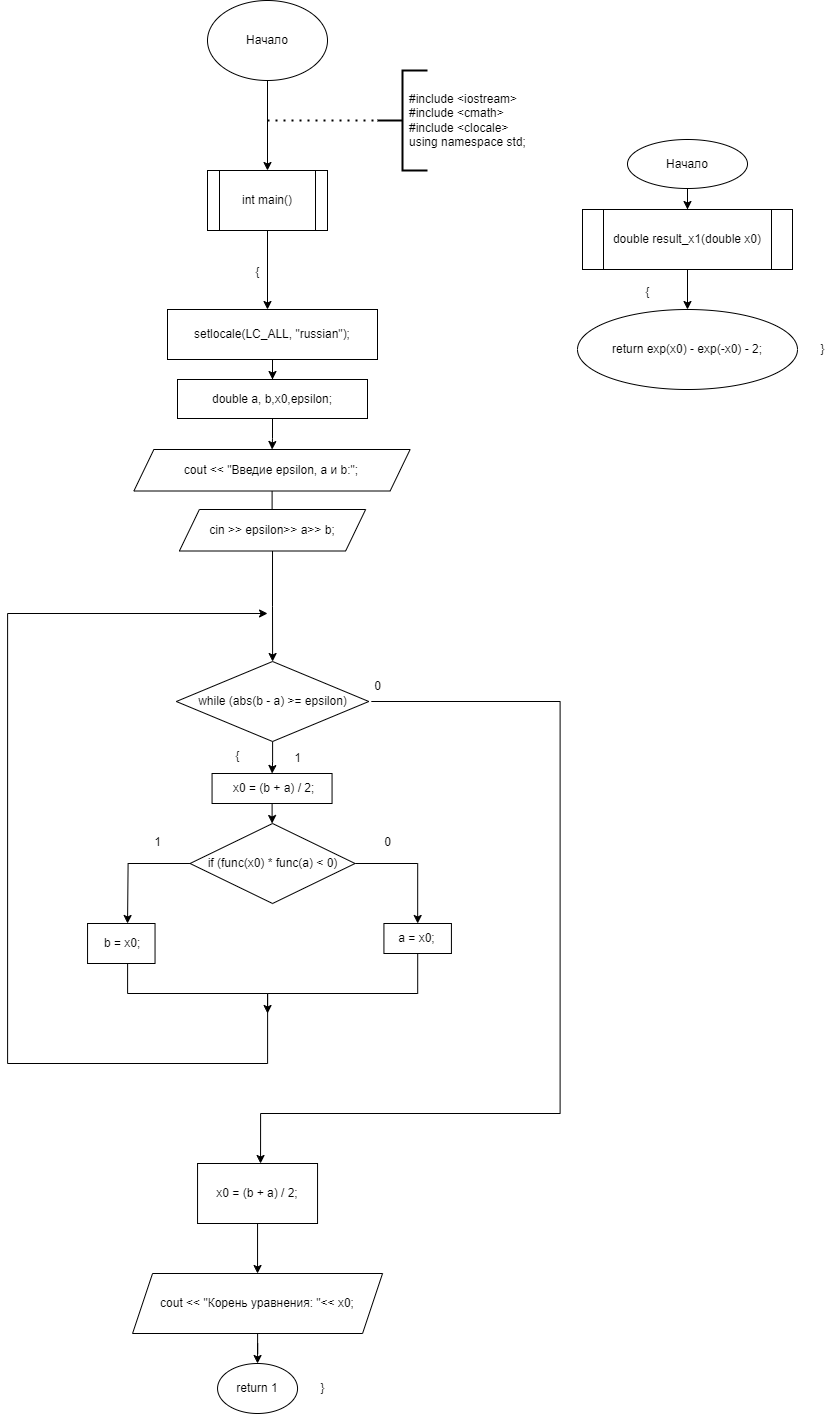
Сначала выбираем начальное приближение, деля отрезок пополам, т.е.

*х0 = (a+b)/2.*

Если *F(x)=0*, то x0 является корнем уравнения. Если F(x) 0, то выбираем тот из отрезков, на концах которого функция имеет противоположные знаки. Полученный отрезок снова делим пополам и выполняем действия сначала и т.д.

Процесс деления отрезка продолжаем до тех пор, пока длина отрезка, на концах которого функция имеет противоположные знаки, не будет меньше заданного числа ε.

**Блок-схема без рекурсии:**

**Код без рекурсии:**

#include <iostream>

#include <cmath>

using namespace std;

double func(double x0)

{

return exp(x0) - exp(-x0) - 2;

}

int main()

{

setlocale(LC\_ALL, "russian");

cout << "Введие epsilon, a и b:";

double epsilon, a, b, x0;

cin >> epsilon >> a >> b;

while (abs(b - a) >= epsilon)

{

x0 = (b + a) / 2;

if (func(x0) \* func(a) < 0)

{

b = x0;

}

else

{

a = x0;

}

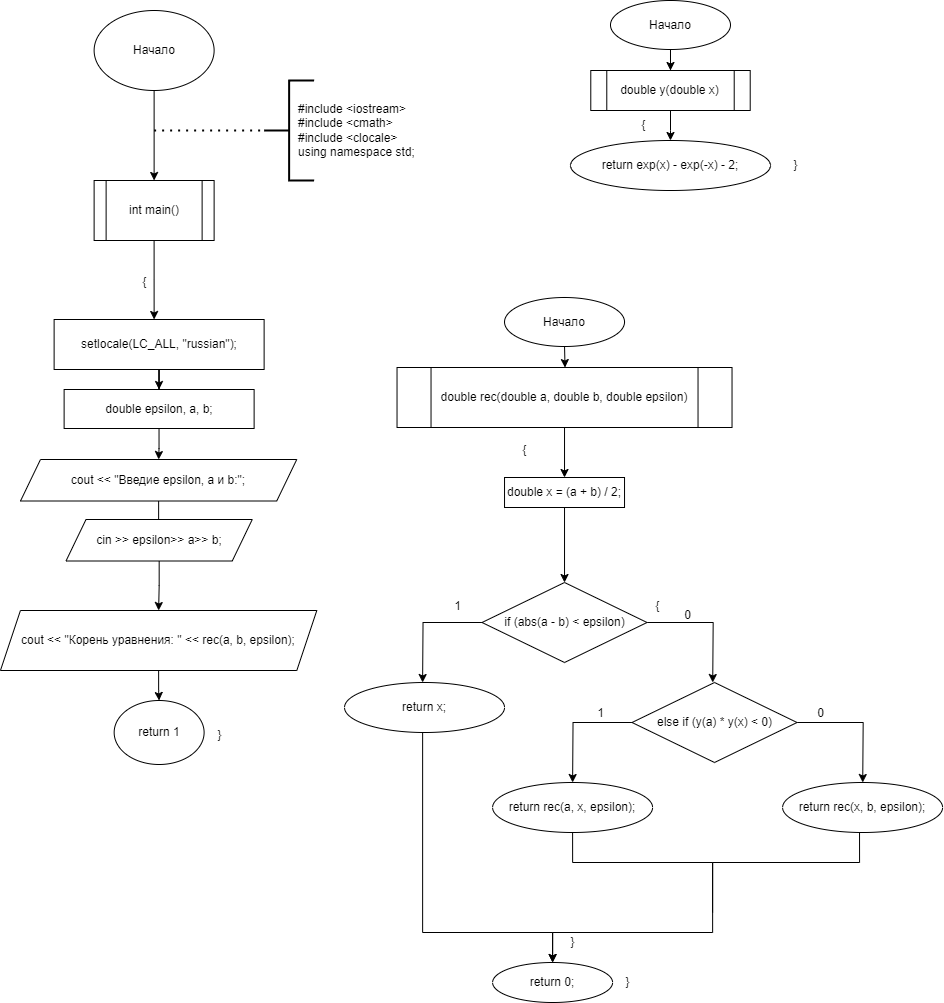
}

x0 = (b + a) / 2;

cout << "Корень уравнения: "<< x0;

}

**Блок-схема c рекурсией:**



**Код с рекурсией:**

#include <iostream>

#include <cmath>

using namespace std;

double y(double x)

{

return exp(x) - exp(-x) - 2;

}

double rec(double a, double b, double epsilon)

{

double x = (a + b) / 2;

if (abs(a - b) < epsilon)

{

return x;

}

else if (y(a) \* y(x) < 0)

{

return rec(a, x, epsilon);

}

else

{

return rec(x, b, epsilon);

}

}

int main()

{

setlocale(LC\_ALL, "russian");

cout << "Введие epsilon, a и b:";

double epsilon, a, b;

cin >> epsilon >> a >> b;

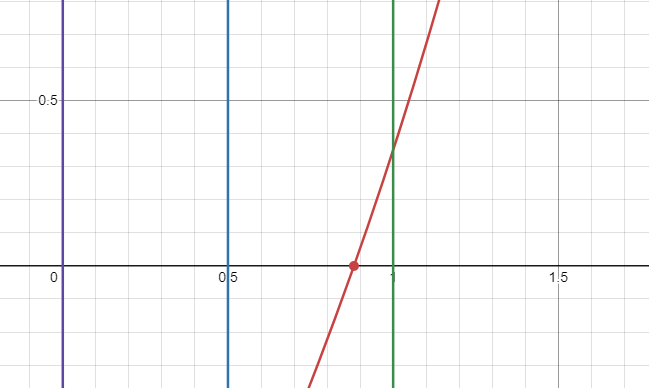
cout << "Корень уравнения: " << rec(a, b, epsilon);

}

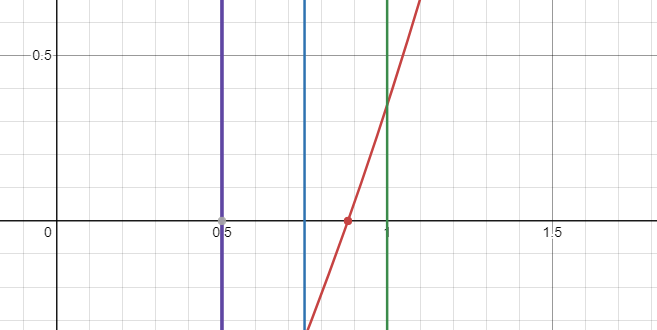
**Пошаговые результаты:**

F(x)-красный, а-фиолетовый, b-зелёный,х0-синий

a=0 b=1 x0=0.5



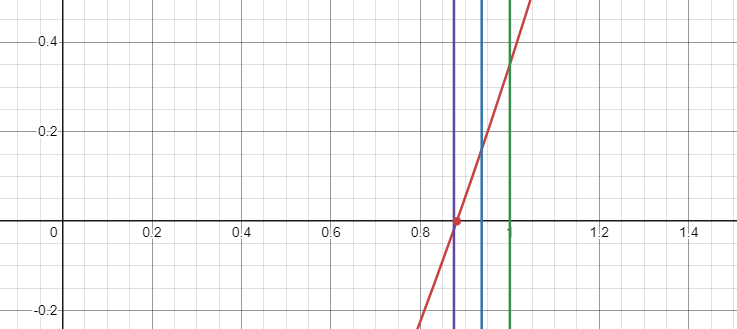
a=0.5 b=1 x0=0.75

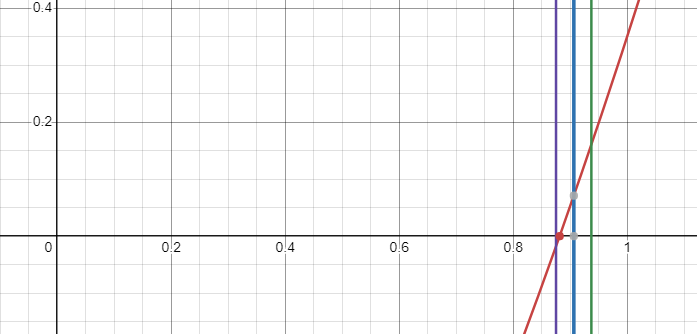


a=0.75 b=1 x0=0.875



a=0.875 b=1 x0=0.9375



a=0.875 b=0.9375 x0=0.90625

**Загрузка файлов на github:**

